

Interval-based Synthesis

Angelo Montanari

Department of Mathematics and Computer Science
University of Udine, Italy
angelo.montanari@uniud.it

Pietro Sala

Department of Computer Science
University of Verona, Italy
pietro.sala@univr.it

In this paper, we introduce the synthesis problem for Halpern and Shoham’s interval temporal logic [5] extended with an equivalence relation \sim over time points ($HS \sim$ for short). In analogy to the case of monadic second-order logic of one successor [2], given an $HS \sim$ formula φ and a finite set Σ_{\square}^T of proposition letters and temporal requests, the problem consists of establishing whether or not, for all possible evaluations of elements in Σ_{\square}^T in every interval structure, there is an evaluation of the remaining proposition letters and temporal requests such that the resulting structure is a model for φ . We focus our attention on the decidability of the synthesis problem for some meaningful fragments of $HS \sim$, whose modalities are drawn from $\{A \text{ (meets)}, \bar{A} \text{ (met by)}, B \text{ (begun by)}, \bar{B} \text{ (begins)}\}$, interpreted over finite linear orders and natural numbers. We prove that the synthesis problem for $AB\bar{B} \sim$ over finite linear orders is decidable (non-primitive recursive hard), while $A\bar{A}B\bar{B}$ turns out to be undecidable. In addition, we show that if we replace finite linear orders by natural numbers, then the problem becomes undecidable even for $AB\bar{B}$.

1 Introduction

Since its original formulation by Church [3], the synthesis problem has received a lot of attention in the computer science literature. A solution to the problem was provided by Büchi and Landweber in [2]. In the last years, a number of extensions and variants of the problem have been investigated, e.g., [12, 13]. The synthesis problem for (point-based) temporal logic has been addressed in [4, 6, 11].

In this paper, we formally state the synthesis problem for interval temporal logic and present some basic results about it. We restrict ourselves to some meaningful fragments of Halpern and Shoham’s modal logic of time intervals [5] extended with an equivalence relation \sim over time points ($HS \sim$ for short). The emerging picture is quite different from the one for the classical synthesis problem (for MSO). In [12], Rabinovich proves that the decidability of the monadic second-order theory of one successor $MSO(\omega, <)$ extended with a unary predicate P ($MSO(\omega, <, P)$ for short) entails the decidability of its synthesis problem, that is, the synthesis problem for a monadic second-order theory is decidable if and only if its underlying theory is decidable. Here, we show that this is not the case with interval temporal logic. We focus our attention on two fragments of HS , namely, the logic $AB\bar{B}$ of Allen’s relations *meets*, *begun by*, and *begins*, and the logic $A\bar{A}B\bar{B}$ obtained from $AB\bar{B}$ by adding a modality for the Allen relation *met by*. In [9], Montanari et al. showed that the satisfiability problem for $AB\bar{B}$ over finite linear orders and the natural numbers is *EXPSpace-complete*, while, in [8], Montanari, Puppis, and Sala proved that the satisfiability problem for $A\bar{A}B\bar{B}$ over finite linear orders is decidable, but not primitive recursive (and undecidable over the natural numbers). In this paper, we prove that the synthesis problem for $AB\bar{B}$ over the natural numbers and for $A\bar{A}B\bar{B}$ over finite linear orders turns out to be undecidable. Moreover, we show there is a significant blow up in computational complexity moving from the satisfiability to the synthesis problem for $AB\bar{B}$ over finite linear orders: while the former is *EXPSpace-complete*, the latter is *NON-PRIMITIVE RECURSIVE-hard*. As a matter of fact, such an increase in the complexity is paired with an increase in the expressive power of the logic: one can exploit universally quantified

variables, that is, propositional letters under the control of the environment, to constrain the length of intervals in a way which is not allowed by $AB\bar{B}$.

The rest of the paper is organized as follows. In Section 2, we introduce syntax and semantics of the logic $A\bar{A}B\bar{B} \sim$ and its fragments. In Section 3, we define the synthesis problem for interval temporal logic, focusing our attention on the considered fragments. The problem is then systematically investigated in the next two sections, where decidable and undecidable instances are identified (a summary of the results is given in Table 1). Conclusions provide an assessment of the work and outline future research directions.

2 The logic $A\bar{A}B\bar{B} \sim$ and its fragments

In this section, we provide syntax and semantics of the fragments of $HS \sim$ we are interested in. The maximal fragment that we take into consideration is $A\bar{A}B\bar{B} \sim$, which features unary modalities $\langle A \rangle$, $\langle \bar{A} \rangle$, $\langle B \rangle$, and $\langle \bar{B} \rangle$ for Allen's binary ordering relations *meets*, *met by*, *begun by*, and *begins* [1], respectively, plus a special proposition letter \sim , to be interpreted as an equivalence relation. The other relevant fragments are $A\bar{A}B\bar{B}$, $AB\bar{B} \sim$, and $AB\bar{B}$.

Formally, let Σ be a set of proposition letters, with $\sim \in \Sigma$. Formulas of $A\bar{A}B\bar{B} \sim$ are built up from proposition letters in Σ by using Boolean connectives \vee and \neg , and unary modalities from the set $\{\langle A \rangle, \langle \bar{A} \rangle, \langle B \rangle, \langle \bar{B} \rangle\}$. Formulas of the fragments $A\bar{A}B\bar{B}$, $AB\bar{B} \sim$, and $AB\bar{B}$ are defined in a similar way. We will often make use of shorthands like $\varphi_1 \wedge \varphi_2 = \neg(\neg\varphi_1 \vee \neg\varphi_2)$, $[A]\varphi = \neg\langle A \rangle\neg\varphi$, $[B]\varphi = \neg\langle B \rangle\neg\varphi$, $\text{true} = a \vee \neg a$, and $\text{false} = a \wedge \neg a$, for some $a \in \Sigma$.

As for the semantics, let $\mathbb{D} = (D, <)$ be a linear order, called *temporal domain*. We denote by $\mathbb{I}_{\mathbb{D}}$ the set of all closed intervals $[x, y]$ over \mathbb{D} , with $x = y$ or $x < y$, abbreviated $x \leq y$ (non-strict semantics). We call *interval structure* any Kripke structure of the form $\mathbf{M} = (\mathbb{D}, A, \bar{A}, B, \bar{B}, \mathcal{V})$. $\mathcal{V} : \mathbb{I}_{\mathbb{D}} \rightarrow \mathcal{P}(\Sigma)$ is a function mapping intervals to sets of proposition letters. A, \bar{A}, B , and \bar{B} denote Allen's relations "meet", "met by", "begun by", and "begins", respectively, and are defined as follows: $[x, y] A [x', y']$ iff $y = x'$, $[x, y] \bar{A} [x', y']$ iff $x = y'$, $[x, y] B [x', y']$ iff $x = x' \wedge y' < y$, and $[x, y] \bar{B} [x', y']$ iff $x = x' \wedge y < y'$. For the sake of brevity, in the following we will write $\mathbf{M} = (\mathbb{D}, \mathcal{V})$ for $\mathbf{M} = (\mathbb{D}, A, \bar{A}, B, \bar{B}, \mathcal{V})$.

Formulas are interpreted over an interval structure $\mathbf{M} = (\mathbb{D}, \mathcal{V})$ and an initial interval $I \in \mathbb{I}_{\mathbb{D}}$ as follows: $\mathbf{M}, I \models a$ iff $a \in \mathcal{V}(I)$, $\mathbf{M}, I \models \neg\varphi$ iff $\mathbf{M}, I \not\models \varphi$, $\mathbf{M}, I \models \varphi_1 \vee \varphi_2$ iff $\mathbf{M}, I \models \varphi_1$ or $\mathbf{M}, I \models \varphi_2$, and, for all $R \in \{A, \bar{A}, B, \bar{B}\}$,

$$\mathbf{M}, I \models \langle R \rangle \varphi \quad \text{iff} \quad \text{there exists } J \in \mathbb{I}_{\mathbb{D}} \text{ such that } I R J \text{ and } \mathbf{M}, J \models \varphi.$$

The special proposition letter \sim is interpreted as an equivalence relation over \mathbb{D} , that is, (i) $x \sim x$ for all $x \in D$, (ii) for all $x, y \in D$, if $x \sim y$, then $y \sim x$, and for all $x, y, z \in D$, if $x \sim y$ and $y \sim z$, then $x \sim z$. Now, for all $x, y \in D$, with $x \leq y$, $\mathbf{M}, [x, y] \models \sim$ if (and only if) $x \sim y$. In the following, we will write $x \sim y$ for $\mathbf{M}, [x, y] \models \sim$ whenever the context, that is, the pair $(\mathbf{M}, [x, y])$, is not ambiguous.

We say that a formula φ is *satisfiable* over a class \mathcal{C} of interval structures if $\mathbf{M}, I \models \varphi$ for some $\mathbf{M} = (\mathbb{D}, \mathcal{V})$ in \mathcal{C} and some interval $I \in \mathbb{I}_{\mathbb{D}}$. In the following, we restrict our attention to the class \mathcal{C}_{fin} of finite linear orders and to (the class $\mathcal{C}_{\mathbb{N}}$ of linear orders isomorphic to) \mathbb{N} . Without loss of generality (we can always suitably rewrite φ), we assume the initial interval on which φ holds (in a model for it) to be the interval $[0, 0]$.

In the following, we will often make use of the following formulas. The formula $[B]\text{false}$ (hereafter, abbreviated π) holds over all and only the *singleton* intervals $[x, x]$. Similarly, the formula $[B][B]\text{false}$ (abbreviated *unit*) holds over the *unit-length* intervals over a discrete order, e.g., over the intervals of \mathbb{N}

of the form $[x, x + 1]$. Finally, the formula $[A][A]\varphi$ ($[G]\varphi$ for short), interpreted over the initial interval $[0, 0]$, forces φ to hold universally, that is, over all intervals. For the sake of readability, from now on, we will denote by Σ the set of all and only those proposition letters that appear in the formula φ under consideration, thus avoiding tedious parametrization like $\Sigma(\varphi)$ (it immediately follows that Σ is always assumed to be a finite set of proposition letters).

Given an $A\bar{A}B\bar{B} \sim$ formula φ , we define its *closure* as the set $\text{closure}(\varphi)$ of all its sub-formulas and all their negations (we identify $\neg\neg\psi$ with ψ , $\neg\langle A \rangle\psi$ with $[A]\neg\psi$, and so on). For a technical reason that will be clear soon, we also introduce the *extended closure* of φ , denoted by $\text{closure}^+(\varphi)$, that extends $\text{closure}(\varphi)$ by adding all formulas of the form $\langle R \rangle\psi$ and $[R]\psi$, for $R \in \{A, B, \bar{A}, \bar{B}\}$ and $\psi \in \text{closure}(\varphi)$. Moreover, we denote by $\text{TF}^+(\varphi) \subseteq \text{closure}^+(\varphi)$ the set $\{\langle R \rangle\psi \in \text{closure}^+(\varphi) : R \in \{A, \bar{A}, B, \bar{B}\}\}$. From now on, given an $A\bar{A}B\bar{B} \sim$ formula φ , we will denote by Σ^T the set $\Sigma \cup \text{TF}^+(\varphi)$.

3 The synthesis problem for interval temporal logic

We are now ready to define the synthesis problem for the interval logic $A\bar{A}B\bar{B} \sim$ (the definition immediately transfers to all its fragments) with respect to the class of finite linear orders and to (any linear order isomorphic to) \mathbb{N} . Without loss of generality, we will refer to a linear order which is either \mathbb{N} or one of its finite prefixes. To start with, we introduce the notion of admissible run.

Definition 1. *Let φ be an $A\bar{A}B\bar{B} \sim$ formula and let $\Sigma_{\square}^T \subseteq \Sigma^T$. An admissible run ρ on the pair $(\varphi, \Sigma_{\square}^T)$ is a finite or infinite sequence of pairs $\rho = ([x_0, y_0], \sigma_0)([x_1, y_1], \sigma_1) \dots$ such that:*

1. *if ρ is finite, that is, $\rho = ([x_0, y_0], \sigma_0) \dots ([x_n, y_n], \sigma_n)$, then there exists $m > 0$ such that $n = 2 \cdot m \cdot (m + 1)$ and, for each $[x, y] \in \mathbb{I}(\{0, \dots, m - 1\})$, there exists $0 \leq i \leq n$ such that $[x, y] = [x_i, y_i]$, while if ρ is infinite, then, for every $[x, y] \in \mathbb{I}(\mathbb{N})$, there exists $i \geq 0$ such that $[x, y] = [x_i, y_i]$;*
2. *$[x_0, y_0] = [0, 0]$, for all $0 < i (\leq n)$, $[x_i, y_i] \in \mathbb{I}(\mathbb{N})$, and for every even index i , $[x_i, y_i] = [x_{i+1}, y_{i+1}]$, $\sigma_i \subseteq \Sigma_{\square}^T$ (the set of proposition letters and temporal requests in Σ_{\square}^T true on $[x_i, y_i]$), $\sigma_{i+1} \subseteq \Sigma^T \setminus \Sigma_{\square}^T$ (the set of proposition letters and temporal requests in $\Sigma^T \setminus \Sigma_{\square}^T$ true on $[x_i, y_i]$), and for all j , with $j \neq i$ and $j \neq i + 1$, $[x_i, y_i] \neq [x_j, y_j]$;*
3. *if $y_{i+1} \neq y_i$, then $y_{i+1} = y_i + 1$ and for all $[x, y] \in \mathbb{I}(\mathbb{N})$, with $y < y_{i+1}$, there exists $0 \leq j < i + 1$ such that $[x_j, y_j] = [x, y]$.*

Conditions 1-3 define the rules of a possibly infinite game between two players \square (spoiler) and \diamond (duplicator), which are responsible of the truth values of proposition letters and temporal requests in Σ_{\square}^T and $\Sigma^T \setminus \Sigma_{\square}^T$, respectively. The game can be informally described as follows. At the beginning, \square chooses an interval $[x, y]$ and defines his labeling for $[x, y]$; \diamond replies to \square by defining her labeling for $[x, y]$ as required by condition 2. In general, \square makes his moves at all even indexes, while \diamond executes her moves at all odd indexes by completing the labeling of the interval chosen by \square at the previous step. Condition 1 guarantees that all intervals on \mathbb{N} (infinite case) or on a finite prefix of it (finite case) are visited by the play. Condition 2 forces every visited interval to be visited exactly once. Condition 3 imposes an order according to which intervals are visited. More precisely, condition 3 prevents \square from choosing an interval $[x, y]$ before he has visited all intervals $[x'y']$, with $y' < y$, that is, \square cannot jump ahead along the time domain without first defining the labeling of all intervals ending at the points he would like to cross.

Let $\text{runs}(\varphi, \Sigma_{\square}^T)$ be the set of all possible admissible runs on the pair $(\varphi, \Sigma_{\square}^T)$. We denote by $\square\text{-pre}(\varphi, \Sigma_{\square}^T)$ the set of all odd-length finite prefixes of admissible runs in $\text{runs}(\varphi, \Sigma_{\square}^T)$, that is, prefixes in which the last move was done by \square , and by $\square\text{-proj}(\varphi, \Sigma_{\square}^T)$ the set of all infinite subsequences of admissible runs in $\text{runs}(\varphi, \Sigma_{\square}^T)$ that contain all and only the pairs occurring at even positions (formally,

$\square\text{-proj}(\varphi, \Sigma_{\square}^T) = \{\rho' = ([x_0, y_0], \sigma_0)([x_1, y_1], \sigma_2) \dots : \exists \rho \in \text{runs}(\varphi, \Sigma_{\square}^T) \text{ such that } \forall i(\rho[2i] = \rho'[i])\}$. It can be easily seen that an admissible run ρ provides a labeling \mathcal{V} for some candidate model of φ by enumerating all its intervals $[x, y]$ following the order of their right endpoints y , that is, for any point y , intervals of the form $[x, y]$ may appear in ρ shuffled in an arbitrary order, which depends on the choices of \square , but if ρ features a labelled interval $[x, y+1]$, then all labeled intervals $[x', y]$, with $0 \leq x' \leq y$, must occur in ρ before it. For any $\rho = ([x_0, y_0], \sigma_0)([x_1, y_1], \sigma_1) \dots$ in $\text{runs}(\varphi, \Sigma_{\square}^T)$, we denote by ρ_I and by ρ_{σ} the sequence $[x_0, y_0][x_1, y_1] \dots$ and the sequence $\sigma_0 \sigma_1 \dots$ obtained by projecting ρ on its first component and its second component, respectively.

Any admissible run ρ on $(\varphi, \Sigma_{\square}^T)$ induces an interval structure $\mathbf{M}_{\rho} = (\mathbb{D}, \mathcal{V})$, called *induced structure*, where $\mathbb{D} = \mathbb{N}$, if ρ is infinite, or $\mathbb{D} = \{0 < \dots < m-1\}$, with $|\rho| = 2 \cdot m \cdot (m+1)$ (such an m exists by definition of finite admissible run) otherwise, and $\mathcal{V}([x, y]) = (\rho[i] \cap \Sigma) \cup (\rho[i+1] \cap \Sigma)$, where i is even and $\rho_I[i] = [x, y]$. Both the existence and the uniqueness of such an index i are guaranteed by condition 2 of the definition of admissible run, and thus the function \mathcal{V} is correctly defined. In particular, we can define a (unique) bijection $f_{\rho} : \mathbb{I}(\mathbb{D}) \rightarrow \mathbb{N}$ such that, for each $[x, y] \in \mathbb{I}(\mathbb{D})$, $f_{\rho}([x, y])$ is even and $\rho_I[f_{\rho}([x, y])] = [x, y]$.

We say that an admissible run ρ is *successful* if and only if $\mathbf{M}_{\rho} = (\mathbb{D}, \mathcal{V}), [0, 0] \models \varphi$ and for all $[x, y] \in \mathbb{I}(\mathbb{D})$ and $\psi \in \text{TF}^+(\varphi)$, it holds that $\mathbf{M}_{\rho}, [x, y] \models \psi$ iff $\psi \in \rho_{\sigma}[f_{\rho}([x, y])] \cup \rho_{\sigma}[f_{\rho}([x, y]) + 1]$. Given a pair $(\varphi, \Sigma_{\square}^T)$, a Σ_{\square}^T -response strategy is a function $S_{\diamond} : \square\text{-pre}(\varphi, \Sigma_{\square}^T) \rightarrow \mathcal{P}(\Sigma^T \setminus \Sigma_{\square}^T)$. Moreover, given an infinite Σ_{\square}^T -response strategy S_{\diamond} and a sequence $\rho_{\square} = ([x_0, y_0], \sigma_0)([x_1, y_1], \sigma_1) \dots$ in $\square\text{-proj}(\varphi, \Sigma_{\square}^T)$, we define the *response of S_{\diamond} to ρ_{\square}* as the infinite admissible run $\rho = ([x'_0, y'_0], \sigma'_0)([x'_1, y'_1], \sigma'_1) \dots$, where, for all $i \in \mathbb{N}$, $\rho[2i] = \rho_{\square}[i]$ and $\rho[2i+1] = (\rho_I[2i], S_{\diamond}(\rho[0 \dots 2i]))$.

The *finite synthesis* problem for $A\bar{A}B\bar{B} \sim$, that is, the winning condition for \diamond on the game defined by conditions 1-3, can be formulated as follows.

Definition 2. *Let φ be an $A\bar{A}B\bar{B} \sim$ formula and $\Sigma_{\square}^T \subseteq \Sigma^T$. We say that the pair $(\varphi, \Sigma_{\square}^T)$ admits a finite synthesis if and only if there exists a Σ_{\square}^T -response strategy S_{\diamond} such that for every $\rho_{\square} \in \square\text{-proj}(\varphi, \Sigma_{\square}^T)$, the response ρ of S_{\diamond} to ρ_{\square} has a finite prefix $\rho[0 \dots n]$ which is a successful admissible run.*

By definition of (finite) admissible run, there exists m such that $n = 2 \cdot m \cdot (m+1)$. Basically, when the labeling is completed for the intervals ending at some point $y' \leq y$ and $\mathbf{M}_{\rho[0 \dots 2 \cdot (y+1) \cdot (y+2)]}$ is a model for φ , then \diamond wins and she can safely ignore the rest of the run ρ (as it happens with reachability games). To generalize the above definition to the \mathbb{N} -synthesis problem, it suffices to drop the prefix condition of Definition 2 and to constrain ρ to be a successful admissible run. In general, we say that a pair $(\varphi, \Sigma_{\square}^T)$ is a *positive instance* of the finite synthesis (resp., \mathbb{N} -synthesis) problem if and only if it admits a finite synthesis (resp., \mathbb{N} -synthesis). A Σ_{\square}^T -response strategy S_{\diamond} , which witnesses that $(\varphi, \Sigma_{\square}^T)$ is a positive instance of the finite synthesis (resp., \mathbb{N} -synthesis) problem, is called a *winning strategy*.

We conclude the section by showing how to exploit the finite synthesis problem to express in $A\bar{B}\bar{B}$ (the smallest fragment we consider in this work) a temporal property that can be expressed neither in $A\bar{B}\bar{B}$ nor in $A\bar{A}B\bar{B}$ (in the usual satisfiability setting). While there is a common understanding of what is meant by enforcing a property on a model via satisfiability, such a notion has various interpretations in the synthesis framework. We assume the following interpretation: forcing a property P on a model means requiring that for all Σ_{\square}^T -response winning strategies S_{\diamond} , there exists a sequence $\rho_{\square} \in \square\text{-proj}(\varphi, \Sigma_{\square}^T)$ such that the run ρ , which is the response of S_{\diamond} to ρ_{\square} , features the property P on all possible models $\mathbf{M}_{\rho[0 \dots n]}$, where $\rho[0 \dots n]$ is a successful admissible run. This amounts to say that no matter how \diamond plays, if she wants to win, then there is always a choice for \square that constrains property P to hold on the model that is built at the end of the play (that is, \diamond cannot win avoiding property P).

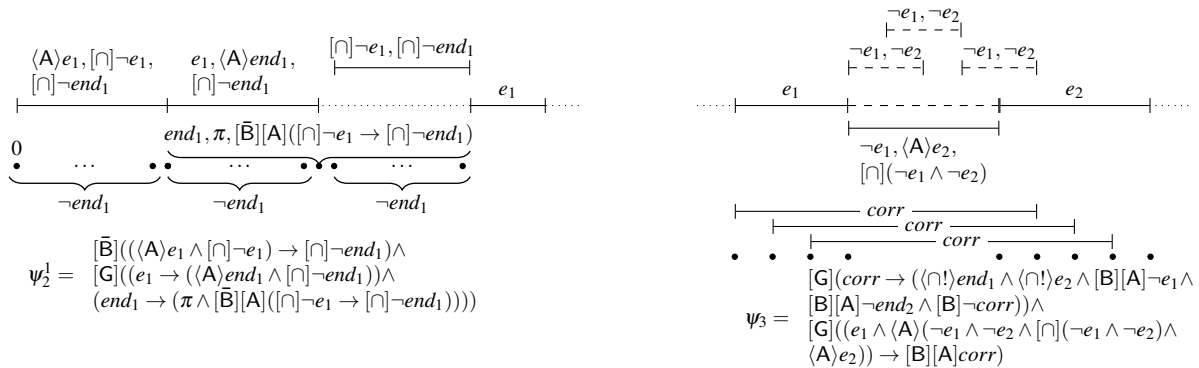
Let us consider, for instance, the following property: *there exists at least one occurrence of an event (of type) e_1 , each occurrence of e_1 is followed by an occurrence of an event (of type) e_2 , occurrences of*

e_1 are disjoint, occurrences of e_2 are disjoint, occurrences of e_1 and e_2 are disjoint, and for every two consecutive occurrences of e_1 and e_2 , the duration of the occurrence of e_2 is greater than or equal to the duration of the occurrence of e_1 .

In the following, we specify the input $(\varphi, \{corr_{\square}\})$ of a synthesis problem, where φ is defined as the conjunction $\psi_0 \wedge \psi_1 \wedge \psi_2^1 \wedge \psi_2^2 \wedge \psi_3 \wedge \psi_4$ and \square controls the proposition letter $corr_{\square}$ only. To simplify the encoding, we will make use two auxiliary modalities $[\cap]\psi = [B][A]\psi \wedge [B]\psi$ and $\langle \cap! \rangle\psi = [B]\neg\psi \wedge \langle B \rangle \langle A \rangle \psi \wedge [B](\langle A \rangle \psi \rightarrow [B][A]\neg\psi)$. By definition, $[\cap]\psi$ holds on an interval $[x, y]$ if ψ holds on all intervals beginning at some z , with $x \leq z < y$, and different from $[x, y]$, while $\langle \cap! \rangle\psi$ holds on $[x, y]$ if there exists one and only one interval beginning at some z , with $x < z < y$, on which ψ holds. For the sake of simplicity, we constrain e_1 and e_2 to hold only over intervals with a duration by means of the formula $[G]((e_1 \vee e_2) \rightarrow \neg\pi)$ (formula ψ_0).

Formula ψ_1 takes care of the initial condition (there exists at least one occurrence of e_1) and of the relationships between e_1 - and e_2 -labeled intervals. Its first conjunct forces the first event to be e_1 . The second one (whose outermost operator is $[G]$) constrains events e_1 to be pairwise disjoint and disjoint from events e_2 , forces each e_1 -labeled interval to be followed by an e_2 -labeled one, and constrains events e_2 to be pairwise disjoint and disjoint from events e_1 .

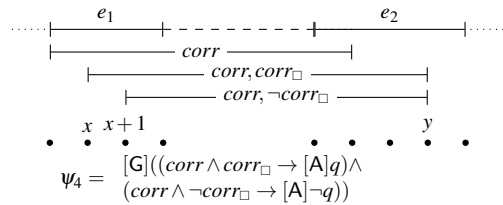
Formula ψ_2^1 forces an auxiliary proposition letter end_1 to hold only at the right endpoint of e_1 -labeled intervals. The first conjunct forces $\neg end_1$ to hold on all intervals that precedes the first occurrence of an e_1 -labeled interval. The second one (whose outermost operator is $[G]$) forces every e_1 -labeled interval to meet an end_1 -labeled interval and prevents end_1 labeled-intervals from occurring inside an e_1 -labeled interval. Moreover, it forces end_1 to hold on point intervals only and constrains all point-intervals (but the first one) that belong to an interval that connects two consecutive e_1 -labeled intervals to satisfy $\neg end_1$. Formula ψ_2^2 imposes the very same conditions on proposition letter end_2 with respect to e_2 -labelled intervals, and it can be obtained from ψ_2^1 by replacing end_1 by end_2 and e_1 by e_2 .



Formula ψ_3 makes use of the proposition letter $corr$ to establish a correspondence between consecutive e_1 - and e_2 -labeled intervals, that is, $corr$ maps points belonging to an e_1 -labeled interval $[x, y]$ to points belonging to an e_2 -labeled $[x', y']$ if and only if there is no point $y < y'' < y'$ that begins an e_1 - or an e_2 -labeled interval. The first conjunct (whose outermost operator is $[G]$) constrains every $corr$ -

labeled interval to cross exactly one point labeled with end_1 and to include the starting point of exactly one e_2 -labeled interval. Moreover, it prevents e_1 -labeled intervals to begin and e_2 -labeled interval to end at a point belonging to a $corr$ -labeled interval. Finally, it allows at most one $corr$ -labeled interval to start at any given point. The second conjunct (whose outermost operator is $[G]$) forces a $corr$ -labeled interval to start at any point belonging to an e_1 -labeled interval $[x, y]$ which has an e_2 -labeled interval as its next e_i -labeled interval, with $i \in \{1, 2\}$. All in all, it constrains $corr$ -labeled intervals to connect all points belonging to the e_1 -labeled interval to the points belonging to the next e_2 -labeled interval. Since exactly one $corr$ -labeled interval can start at each point in e_1 , $corr$ can be viewed as a function from points in e_1 to points in e_2 . To capture the intended property, however, we further need to force such a function to be injective. This is done by formula ψ_4 , which exploits the interplay between \square and \diamond .

It is worth pointing out that if we take a look at formula ψ_4 from the point of view of the satisfiability problem, it does not add any constraint to the proposition letter $corr$. Indeed, ψ_4 can be trivially satisfied by forcing $corr_\square$ and q to be always true or always false in the model. In the context of the (finite) synthesis problem, things are different: \diamond has no control on the proposition letter $corr_\square$, and if she tries to violate injectivity of $corr$ (as depicted in the graphical account for ψ_4), then \square has a strategy to win, as shown by the following (portion of a) run:



$$\psi_4 = [G]((corr \wedge corr_\square \rightarrow [A]q) \wedge (corr \wedge \neg corr_\square \rightarrow [A]\neg q))$$

$$\dots ([x, y], \{corr_\square\}) ([x, y], \{\dots, corr, \dots\}) ([x+1, y], \emptyset) ([x+1, y], \{\dots, corr, \dots\}) ([y, y], \{\dots\}) ([y, y], \{\dots\}) \dots$$

In general, \square has a strategy to impose that for each point y , there exists at most one $corr$ -labeled interval. Suppose that, at a certain position of the run, ρ \square and \diamond are playing on the labeling of all intervals ending at a given point y . For each $0 \leq x \leq y$, \square , who always plays first, may choose any value for $corr_\square$ on $[x, y]$ until \diamond chooses to put $corr$ in her reply to a move of \square on an interval $[x', y]$, for some x' . From that point on, for all intervals $[x'', y]$ which have not been labeled yet, if \square has put $corr_\square$ on $[x', y]$, he will label $[x'', y]$ with $\neg corr_\square$, and if he has put $\neg corr_\square$ on $[x', y]$, he will label $[x'', y]$ with $corr_\square$.

4 The big picture: undecidability and complexity reductions

In this section, we state (un)decidability and complexity results for the synthesis problem for the three fragments $AB\bar{B}$, $AB\bar{B} \sim$, and $A\bar{A}B\bar{B}$. We consider both finite linear orders and the natural numbers. The outcomes of such an analysis are summarized in Table 1. All the reductions we are going to define make use of (Minsky) counter machines or of their lossy variants. A (Minsky) counter machine [7] is a triple $\mathcal{M} = (Q, k, \delta)$, where Q is a finite set of states, k is the number of counters, whose values range over \mathbb{N} , and δ is a function that maps each state $q \in Q$ to a transition rule having one of the following forms:

- $inc(i)$ and $goto(q')$, where $i \in \{1, \dots, k\}$ is a counter and $q' \in Q$ is a state: whenever \mathcal{M} is in state q , then it first increments the value of counter i and then it moves to state q' ;
- if $i=0$ then $goto(q')$ else $dec(i)$ and $goto(q'')$, where $i \in \{1, \dots, k\}$ is a counter and $q', q'' \in Q$ are states: whenever \mathcal{M} is in state q and the value of the counter i is equal to 0 (resp., greater than 0), then \mathcal{M} moves to state q' (resp., it decrements the value of i and moves to state q'').

A computation of \mathcal{M} is any sequence of configurations that conforms to the semantics of the transition relation. In the following, we will define and exploit a suitable reduction from the problem of deciding,

Logic	Linear Order	Satisfiability	Synthesis
$AB\bar{B}$	Finite	Decidable [9] (EXPSpace-complete)	Decidable (NonPrimitiveRecursive-hard)
$AB\bar{B}$	\mathbb{N}	Decidable [9] (EXPSpace-complete)	Undecidable
$AB\bar{B} \sim$	Finite	Decidable [10] (NonPrimitiveRecursive-hard)	(NonPrimitiveRecursive-hard)
$AB\bar{B} \sim$	\mathbb{N}	Undecidable [10]	Undecidable
$A\bar{A}B\bar{B}$	Finite	Decidable [8] (NonPrimitiveRecursive-hard)	Undecidable
$A\bar{A}B\bar{B}$	\mathbb{N}	Undecidable [8]	Undecidable
$A\bar{A}B\bar{B} \sim$	Finite	Undecidable [10]	Undecidable
$A\bar{A}B\bar{B} \sim$	\mathbb{N}	Undecidable	Undecidable

Table 1: Decidability and complexity of the satisfiability and synthesis problems for the considered *HS* fragments. Results written in bold are given in the present work; results with no explicit reference immediately follow from those given in this paper or in other referred ones.

given a counter machine $\mathcal{M} = (Q, k, \delta)$ and a pair of control states q_{init} and q_{halt} , whether or not every computation of \mathcal{M} that starts at state q_{init} , with all counters initialized to 0, eventually reaches the state q_{halt} , with the values of all counters equal to 0 (0-0 *reachability* problem).

Theorem 1. ([7]) *The 0-0 reachability problem for counter machines is undecidable.*

If, given a configuration $(q, \bar{z}) \in Q \times \mathbb{N}^k$, we allow a counter machine \mathcal{M} to non-deterministically execute an internal (lossy) transition and to move to a configuration (q, \bar{z}') , with $\bar{z}' \leq \bar{z}$ (the relation \leq is defined component-wise on the values of the counters), we obtain a lossy counter machine.

Theorem 2. ([14]) *The 0-0 reachability problem for lossy counter machines is decidable with NonPrimitive Recursive-hard complexity.*

Notice that, given a counter machine \mathcal{M} , a computation where lossy transitions have been never executed, namely, a *perfect computation*, is a lossy computation, while, in general, a lossy computation cannot be turned into a computation which does not execute any lossy transition.

Since lossy transitions are not under the control of the machine \mathcal{M} and they may take place at each state of the computation, lossy computations and perfect computations can be viewed as two particular semantics for the computations of the same machine, the former being more relaxed (that is, it allows, in general, a larger number of successful computations) than the latter. Now we prove that the synthesis problem for

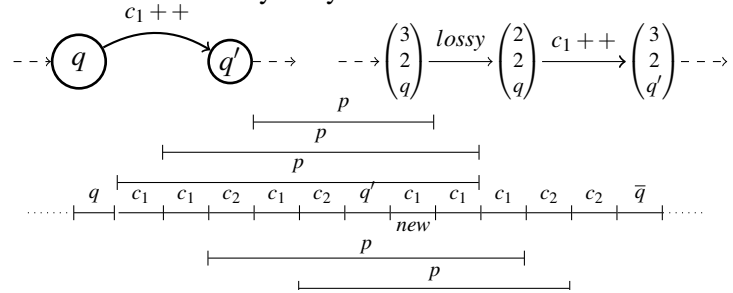


Figure 1: Encoding of a lossy computation in $A\bar{A}B\bar{B}$: incrementing states.

$A\bar{A}B\bar{B}$ over finite linear orders is undecidable. We elaborate on a result given by Montanari et al. in [8], where, for any counter machine \mathcal{M} , a formula $\varphi_{\mathcal{M}}$ is given such that $\varphi_{\mathcal{M}}$ is satisfiable over finite linear orders if and only if the corresponding counter machine \mathcal{M} has a 0-0 lossy computation for two given states q_0 and q_f . The idea is to encode the successful computation in a model for $\varphi_{\mathcal{M}}$. In the following, we will first briefly recall the key ingredients of such an encoding; then, we will show how to extend $\varphi_{\mathcal{M}}$ with an additional formula that actually introduces a new constraint only in the finite synthesis setting. We start with a short explanation of how the basic features of a (candidate) model for $\varphi_{\mathcal{M}}$

can be enforced. To help the reader, we provide a graphical account of the technique (interval structure in Figure 1). Each configuration is encoded by means of a sequence of consecutive unit intervals. The first unit interval of any such sequence is labeled with a propositional letter q_i , where q_i is a state of \mathcal{M} . A unary encoding of the values of the counters is then provided by making use of the unit intervals in between (the unit interval labeled with) q_i and the next unit interval labeled with a state of \mathcal{M} , say, q_j . Any such unit interval is labeled by exactly one proposition letter c_i , with $i \in \{1, \dots, k\}$. More precisely, for all $i \in \{1, \dots, k\}$, the value of the counter c_i in the configuration beginning at q_i is given by the number of c_i -labelled unit intervals between q_i and q_j . We show now how to encode the two kinds of transition of \mathcal{M} . Let q be the current state. We first consider the case of increasing transitions of the form $\text{inc}(i)$ and $\text{goto}(q')$. It is easy to write a formula that forces the unit interval labeled with a state of \mathcal{M} next to q to be labeled with q' . It is also easy to force exactly one c_i -labeled unit interval in the next configuration to be labeled with a special proposition letter new , to identify the c_i -labeled interval just introduced to mimic the increment of the counter i . Let us consider now transitions of the form $\text{if } i=0 \text{ then } \text{goto}(q') \text{ else } \text{dec}(i)$ and $\text{goto}(q'')$. We first verify whether there are not c_i -labeled unit intervals in the current configuration by checking if the formula $[B][A]\neg c_i$ holds over the interval that begins at the left endpoint of (the unit interval labeled with) q and ends at the left endpoint of the next unit interval labeled with a state of \mathcal{M} . If such a formula does not hold, we have to mimic the decreasing of the counter c_i by one. To this end, we introduce another special proposition letter del and we force it to hold over one of the c_i -labeled intervals of the current configuration. Intuitively, the interval marked by del is not transferred to the next configuration, thus simulating the execution of the decrement on the counter c_i . In Figure 3, we graphically depict the encoding of zero-test transitions.

The next step is the correct transfer of all counter values from the current configuration to the next one with the only exception of the new/del -labeled intervals (if any). What does “correctly” mean? According to the definition of the lossy semantics, a counter can be either transferred with its exact value or with a smaller one, that is, we only have to avoid unsupported increments of counter values, as for lower values we can always assume that a lossy transition has been fired (Figure

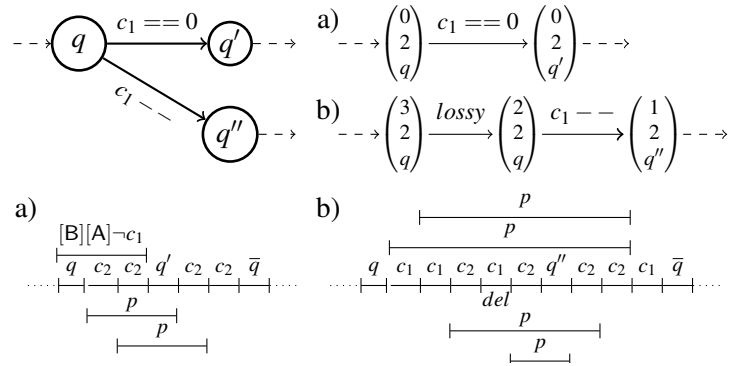


Figure 2: Encoding of a lossy computation in $A\bar{A}B\bar{B}$: zero-test states.

1 gives an example of such a behavior). The transfer is done by means of a function that maps (the left endpoints of) c_i -labeled intervals of the current configuration to (the left endpoints of) c_i -labeled intervals of the next one. Such a function is encoded by a proposition letter p . Notice that all the properties we dealt with so far, including those concerning p , can be expressed in $AB\bar{B}$. However, we still need to suitably constrain p to guarantee that it behaves as expected. In particular, we must impose surjectivity to prevent unsupported increments of counter values from occurring. Such a property can be forced by the formula $\psi_{sur} = [G](\bigwedge_{i \in \{1, \dots, k\}} c_i \wedge \neg new \rightarrow \langle \bar{A} \rangle p)$, where modality $\langle \bar{A} \rangle$ plays an essential role.

Up to this point, the entire encoding has been done without exploiting any special feature of Σ_{\square} and Σ_{\diamond} brought by the finite synthesis context. The power of synthesis is needed to force injectivity, thus turning lossy counter machines into standard ones. Let us define the concrete instance of the synthesis problem we are interested in as the pair $(\varphi_{\mathcal{M}}^{0-0}, \{p_{\square}\})$, where $\varphi_{\mathcal{M}}^{0-0} = \varphi_{\mathcal{M}} \wedge \psi_{inj}$ and $\psi_{inj} = [G]((p \wedge p_{\square} \rightarrow [A](\neg \pi \rightarrow s)) \wedge (p \wedge \neg p_{\square} \rightarrow [A](\neg \pi \rightarrow \neg s)))$ is the formula for injectivity.

Let us take a closer look at $\varphi_{\mathcal{M}}^{0-0}$ to understand how it guarantees injectivity of the transfer function encoded by p . Let us assume $(\varphi_{\mathcal{M}}^{0-0}, \{p_{\square}\})$ to be a positive instance of the finite synthesis problem. Then, there exists a Σ_{\square}^T -response strategy S_{\diamond} such that for every ρ_{\square} , a prefix $\rho[0\dots n]$ of the response ρ of S_{\diamond} to ρ_{\square} is a successful run, that is, $\mathbf{M}_{\rho}[0\dots n]$ satisfies $\varphi_{\mathcal{M}}^{0-0}$. Assume by way of contradiction that for every successful run $\rho[0\dots n]$ (for some natural number n), which is an S_{\diamond} response to ρ_{\square} , the resulting model $\mathbf{M}_{\rho}[0\dots n] = (\{0, \dots, n\}, \mathcal{V})$ is such that there exist three points $x < y < z \leq n$ with $p \in \mathcal{V}(x, z) \cap \mathcal{V}(y, z)$, thus violating injectivity of p . We collect all these triples (x, y, z) into a set $Defects(\rho)$. Without loss of generality, we can assume n to be the minimum natural numbers such that $\mathbf{M}_{\rho}[0\dots n]$ satisfies $\varphi_{\mathcal{M}}^{0-0}$. A lexicographical order \leq can be defined over the triples (x, y, z) in $Defects(\rho)$, where z is considered as the most significant component and x as the least significant one. Let (x, y, z) be the minimum element of $Defects(\rho)$ with respect to \leq . First, we observe that z cannot be equal to n , as, by construction, (i) z is the left endpoint of a counter-labeled interval, (ii) any model for the formula is forced to end at the left endpoint of a state-labeled interval, and (iii) an interval cannot be both a state- and a counter-labeled interval (state- and counter-labeled intervals are mutually exclusive). Hence, $z < n$. Now, since $\mathbf{M}_{\rho}[0\dots n]$ satisfies ψ_{inj} , it immediately follows that either $p_{\square} \in \mathcal{V}(x, z) \cap \mathcal{V}(y, z)$ or $p_{\square} \notin \mathcal{V}(x, z) \cup \mathcal{V}(y, z)$ (otherwise, both $s \in \mathcal{V}(z, z+1)$ and $s \notin \mathcal{V}(z, z+1)$). Without loss of generality, we assume that $p_{\square} \in \mathcal{V}(x, z) \cap \mathcal{V}(y, z)$ (the other case is completely symmetric). Let i, j be the even indexes (player \square is playing at even positions) such that $\rho[i] = ([x, z], \sigma_i)$ and $\rho[j] = ([y, z], \sigma_j)$, respectively. Let $i < j$ (the opposite case is perfectly symmetric). Since φ must be satisfied by all $\rho_{\square} \in \square\text{-proj}(\varphi, \{p_{\square}\})$, there exists a run ρ' such that $\rho'[0\dots j-1] = \rho[0\dots j-1]$ (it is a run where spoiler behaves the same up to position $j-1$, and since S_{\diamond} is a function on the prefixes of runs, duplicator behaves the same as well) and $p_{\square} \notin \rho'[j]_{\sigma}$ (while $p_{\square} \in \rho[j]_{\sigma}$). By hypothesis, there exists n' such that $\rho'[0\dots n']$ is a model for φ . It clearly holds that $n' > z$; otherwise, minimality of n would be violated, since ρ' is equal to ρ up to z . Now, from $p \in \rho'[i+1]_{\sigma}$ and $p_{\square} \in \rho'[i]_{\sigma}$, it follows that $p \notin \rho'[j+1]_{\sigma}$. Otherwise ($p \in \rho'[j+1]_{\sigma}$), a contradiction would occur when point $z+1$ is added, as ψ_{inj} has forced player \diamond to put both s and $\neg s$ on the interval $[z, z+1]$. We can conclude that there exists a run ρ' such that injectivity of p is guaranteed up to the defect (x, y, z) , that is, the minimum element (x', y', z') in $Defects(\rho')$ is greater than (x, y, z) according to the above-defined lexicographical order. Now, we can apply exactly the same argument we use for ρ to ρ' , identifying a run ρ'' whose minimum defect (x'', y'', z'') in $Defects(\rho'')$ is greater than (x', y', z') , and so on. Let ρ_{ω} be the limit run. It holds that ρ_{ω} is still a response of S_{\diamond} to some ρ_{\square} in $\square\text{-proj}(\varphi, \{p_{\square}\})$ where p is injective (contradiction).

Theorem 3. *Let \mathcal{M} be a counter machine. \mathcal{M} is a positive instance of the 0-0 reachability problem if and only if the $A\bar{A}B\bar{B}$ -formula $\varphi_{\mathcal{M}}^{0-0}$ is a positive instance of the finite synthesis problem.*

Corollary 1. *The finite synthesis problem for $A\bar{A}B\bar{B}$ is undecidable.*

As we already pointed out, the modality $\langle \bar{A} \rangle$ comes into play in the specification of surjectivity only. Hence, if we drop the formula φ_{sur} , we can not force surjectivity anymore, but we can still impose injectivity (by exploiting the power of synthesis) in the smaller fragment $AB\bar{B}$. Then, by making use (with minor modifications) of a previous result of ours [10], we can reduce the reachability problem for lossy counter machines to the satisfiability problem for $AB\bar{B} \sim$ over finite linear orders. The main difference from the previous reduction is that computations are encoded backwards, that is, the encoding starts from the final configuration and (following the time line) it reaches the initial one.

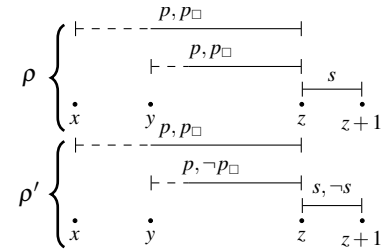


Figure 3: Behavior of the formula ψ_{inj} .

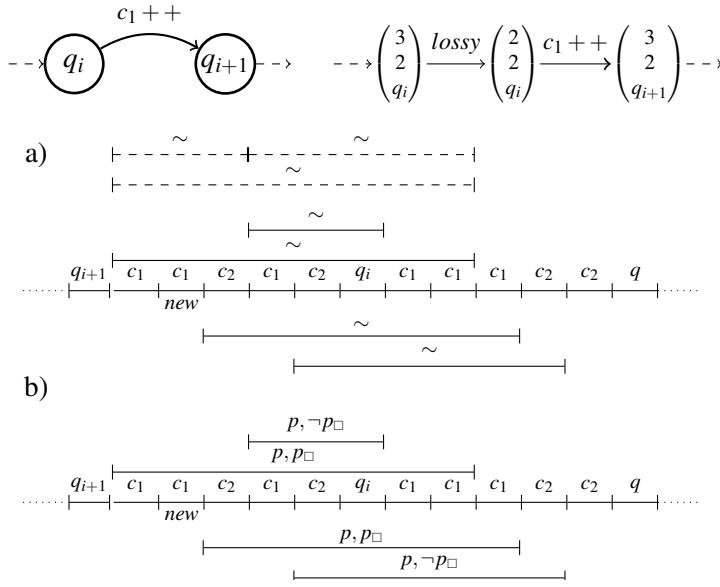


Figure 4: Encoding of a lossy computation in $AB\bar{B} \sim$ satisfiability (a) and $AB\bar{B}$ synthesis (b): incrementing states.

formula $\varphi_{inj}^{\sim} = [G](\sim \wedge \neg \pi) \rightarrow \bigvee_{q \in Q} \langle B \rangle \langle A \rangle q$ states that any \sim -interval, which is not a point-interval, must cross at least one state-interval. It immediately follows that points belonging to the same configuration must belong to different equivalence classes. Suppose that the endpoints of a \sim -interval belong to the same \sim class, against φ_{inj}^{\sim} . Transitivity of \sim can then be exploited to violate injectivity, as shown by the dashed intervals in Figure 4 (a). Injectivity can be forced in $AB\bar{B}$, which does not include the proposition letter \sim , by using the expressive power of synthesis. Thanks to formula ψ_{inj} , we can indeed mimic the behavior of \sim by the combined behavior of p and p_{\square} as shown in Figure 4 (b). As an immediate consequence, the non-primitive recursive hardness of the satisfiability problem for $AB\bar{B} \sim$ over finite linear orders [10] can be directly transferred to the finite synthesis problem for $AB\bar{B}$.

Theorem 4. *The finite synthesis problem for $AB\bar{B}$ is Non-Primitive Recursive hard.*

Similarly, the undecidability of the \mathbb{N} -synthesis problem for $AB\bar{B}$ can be derived from the undecidability of the satisfiability problem of $AB\bar{B} \sim$ over linear orders isomorphic to \mathbb{N} [10].

Theorem 5. *The \mathbb{N} -synthesis problem for $AB\bar{B}$ is undecidable.*

5 Decidability of $AB\bar{B} \sim$ over finite linear orders

We conclude the paper by showing that the synthesis problem for $AB\bar{B} \sim$ over finite linear orders is decidable. To this end, we introduce some basic terminology, notations, and definitions.

Let $\mathbf{M} = (\mathbb{D}, \mathcal{I})$ be an interval structure. We associate with each interval $I \in \mathbb{I}_{\mathbb{D}}$ its φ -type $\text{type}_{\mathbf{M}}^{\varphi}(I)$, defined as the set of all formulas $\psi \in \text{closure}^+(\varphi)$ such that $\mathbf{M}, I \models \psi$ (when no confusion arises, we omit the parameters \mathbf{M} and φ). A particular role will be played by those types F that contains the subformula $[B]\text{false}$, which are necessarily associated with singleton intervals. When no interval structure is given, we can still try to capture the concept of type by means of a maximal “locally consistent” subset of $\text{closure}^+(\varphi)$. Formally, we call φ -atom any set $F \subseteq \text{closure}^+(\varphi)$ such that (i) $\psi \in F$ iff $\neg\psi \notin F$, for all $\psi \in \text{closure}^+(\varphi)$, (ii) $\psi \in F$ iff $\psi_1 \in F$ or $\psi_2 \in F$, for all $\psi = \psi_1 \vee \psi_2 \in \text{closure}^+(\varphi)$, (iii) if $[B]\text{false} \in F$

It can be easily checked that, in the synthesis setting, in order to express the lossy behavior, the transfer function must be injective. The absence of modality $\langle \bar{A} \rangle$ makes it impossible to provide an upper bound to the value of a counter in the next configuration along the time line, which is the previous configuration in the computation of the counter machine. Let us consider Figure 4 (a). The second c_i -labeled interval of the configuration starting at q_i does not begin any \sim -interval. However, since computations are encoded backwards, such a situation can be simulated by introducing a lossy transition. Injectivity can be forced by exploiting the equivalence relation (proposition letter \sim). The formula

and $\psi \in F$, then $\langle A \rangle \psi \in F$, for all $\psi \in \text{closure}(\varphi)$, (iv) if $[B]\text{false} \in F$ and $\langle A \rangle \psi \in F$, then $\psi \in F$ or $\langle \bar{B} \rangle \psi \in F$, for all $\psi \in \text{closure}(\varphi)$, and (v) if $[B]\text{false} \in F$, then $\sim \in F$. We call π -atoms those atoms that contain the formula $[B]\text{false}$, which are thus candidate types for singleton intervals. We denote by $\text{atoms}(\varphi)$ the set of all φ -atoms.

Given an atom F and a relation $R \in \{A, B, \bar{B}\}$, we let $\text{req}_R(F)$ be the set of *requests of F along direction R* , namely, the formulas $\psi \in \text{closure}(\varphi)$ such that $\langle R \rangle \psi \in F$. Similarly, we let $\text{obs}(F)$ be the set of *observables of F* , namely, the formulas $\psi \in F \cap \text{closure}(\varphi)$ – intuitively, the observables of F are those formulas $\psi \in F$ that fulfill requests of the form $\langle R \rangle \psi$ from other atoms. Note that, for all π -atoms F , we have $\text{req}_A(F) = \text{obs}(F) \cup \text{req}_{\bar{B}}(F)$.

It is well known that formulas of interval temporal logics can be equivalently interpreted over the so-called compass structures [15]. These structures can be seen as two-dimensional spaces in which points are labelled with complete logical types (atoms). Such an alternative interpretation exploits the existence of a natural bijection between the intervals $I = [x, y]$ over a temporal domain \mathbb{D} and the points $p = (x, y)$ in the $\mathbb{D} \times \mathbb{D}$ grid such that $x \leq y$. It is useful to introduce a *dummy atom* \emptyset , distinct from all other atoms, and to assume that it labels all and only the points (x, y) such that $x > y$, which do not correspond to intervals. Conventionally, we assume $\text{obs}(\emptyset) = \emptyset$ and $\text{req}_R(\emptyset) = \emptyset$, for $R \in \{A, B, \bar{B}\}$.

Formally, a *compass φ -structure over a linear order \mathbb{D}* is a labeled grid $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$, where the function $\tau : \mathbb{D} \times \mathbb{D} \rightarrow \text{atoms}(\varphi) \uplus \{\emptyset\}$ maps any point (x, y) to either a φ -atom (if $x \leq y$) or the dummy atom \emptyset (if $x > y$). Allen's relations over intervals have analogue relations over points A, B, \bar{B} (by a slight abuse of notation, we use the same letters for the corresponding relations over the points of a compass structure). Thanks to such an interpretation, any interval structure \mathbf{M} can be converted to a compass one $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$ by simply letting $\tau(x, y) = \text{type}([x, y])$ for all $x \leq y \in \mathbb{D}$. The converse, however, is not true in general, as the atoms associated with points in a compass structure may be inconsistent with respect to the underlying geometrical interpretation of Allen's relations. To ease a correspondence between interval and compass structures, we enforce suitable *consistency conditions* on compass structures. First, we constrain each compass structures $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$ to satisfy the following conditions on the special proposition letter \sim (for the sake of readability we write $x \sim y$ in place of $\sim \in \tau(x, y)$): (i) for all $x \in D$, $x \sim x$; (ii) for all $x < y < z$ in D , $x \sim y \wedge y \sim z \rightarrow x \sim z$, $x \sim z \wedge y \sim z \rightarrow x \sim y$, and $x \sim z \wedge x \sim y \rightarrow y \sim z$. Second, to guarantee the consistency of atoms associated with points, we introduce two binary relations over them. Let F and G be two atoms:

$$F \xrightarrow{\bar{B}} G \text{ iff} \quad F \xrightarrow{A} G \text{ iff}$$

$$\begin{cases} \text{req}_{\bar{B}}(F) \supseteq \text{obs}(G) \cup \text{req}_{\bar{B}}(G) \\ \text{req}_B(G) \supseteq \text{obs}(F) \cup \text{req}_B(F) \end{cases} \quad \left\{ \begin{array}{l} \text{req}_A(F) = \text{obs}(G) \cup \text{req}_B(G) \cup \text{req}_{\bar{B}}(G) \end{array} \right.$$

Note that the relation $\xrightarrow{\bar{B}}$ is transitive, while \xrightarrow{A} only satisfies $\xrightarrow{A} \circ \xrightarrow{\bar{B}} \subseteq \xrightarrow{A}$. Observe also that, for all interval structures \mathbf{M} and all intervals I, J in it, if $I \bar{B} J$ (resp., $I A J$), then $\text{type}_{\mathbf{M}}(I) \xrightarrow{\bar{B}} \text{type}_{\mathbf{M}}(J)$ (resp., $\text{type}_{\mathbf{M}}(I) \xrightarrow{A} \text{type}_{\mathbf{M}}(J)$). Hereafter, we tacitly assume that every compass structure $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$ satisfies analogous consistency properties with respect to its atoms, namely, for all points $p = (x, y)$ and $q = (x', y')$ in $\mathbb{D} \times \mathbb{D}$, with $x \leq y$ and $x' \leq y'$, if $p \bar{B} q$ (resp., $p A q$), then $\tau(p) \xrightarrow{\bar{B}} \tau(q)$ (resp., $\tau(p) \xrightarrow{A} \tau(q)$). In addition, we say that a request $\psi \in \text{req}_R(\tau(p))$ of a point p in a compass structure $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$ is *fulfilled* if there is another point q such that $p R q$ and $\psi \in \text{obs}(\tau(q))$ – in this case, we say that q is a *witness of fulfilment of ψ from p* . The compass structure \mathcal{G} is said to be *globally fulfilling* if all requests of all its points are fulfilled.

We can now recall the standard correspondence between interval and compass structures (the proof is based on a simple induction on sub-formulas):

Proposition 1 ([10]). *Let φ be an $\text{AB}\bar{\text{B}} \sim$ formula. For every globally fulfilling compass structure $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$, there is an interval structure $\mathbf{M} = (\mathbb{D}, \mathcal{V})$ such that, for all $x \leq y \in \mathbb{D}$ and all $\psi \in \text{closure}^+(\varphi)$, $\mathbf{M}, [x, y] \models \psi$ iff $\psi \in \tau(x, y)$.*

In view of Proposition 1, the satisfiability problem for an $\text{AB}\bar{\text{B}} \sim$ formula φ reduces to the problem of deciding the existence of a compass $\bar{\varphi}$ -structure $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$, with $\bar{\varphi} = \langle \text{G} \rangle \varphi$ ($\langle \text{G} \rangle \varphi$ is a shorthand for $\neg[\text{G}]\neg\varphi$), that features the observable $\bar{\varphi}$ at every point, that is, $\bar{\varphi} \in \text{obs}(\tau(x, y))$ for all $x \leq y \in \mathbb{D}$.

It can be easily checked that, given an $\text{AB}\bar{\text{B}} \sim$ formula φ , for every set $\sigma \subseteq \Sigma^T$, there exists at most one atom $F \in \text{atoms}(\varphi)$ such that $F \cap \Sigma^T = \sigma$. Hence, for all $\text{AB}\bar{\text{B}} \sim$ formulas φ , we can define a (unique) partial function $f_\varphi : \mathcal{P}(\Sigma^T) \rightarrow \text{atoms}(\varphi)$ such that for every set $\sigma \subseteq \Sigma^T$, $f_\varphi(\sigma) = F$ with $F \cap \Sigma^T = \sigma$. By making use of the function f_φ , given an interval structure $\mathbf{M} = (\mathbb{D}, \mathcal{V})$ for φ , we can define a corresponding compass structure $\mathcal{G}_{\mathbf{M}} = (\mathbb{D} \times \mathbb{D}, \tau)$ such that, for all $[x, y] \in \mathbb{I}(\mathbb{D})$, $\tau(x, y) = f_\varphi(\mathcal{V}([x, y]) \cup \{\psi \in \Sigma^T \cap \text{closure}^+(\varphi)\})$.

Lemma 1. *For every $\text{AB}\bar{\text{B}} \sim$ formula φ and interval structure \mathbf{M} for it, $\mathcal{G}_{\mathbf{M}}$ is a fulfilling compass structure for φ .*

Let φ be an $\text{AB}\bar{\text{B}} \sim$ formula, \mathbf{G}_φ be the set of all finite compass structures for φ , and $\Sigma_\square^T \subseteq \Sigma^T$. We define a Σ_\square^T -response tree as a tuple $\mathcal{T} = (V, E, \mathcal{L}_V, \mathcal{L}_E)$ where

- (V, E) is a finite tree equipped with two labeling functions $\mathcal{L}_V : V \rightarrow \Sigma^T \setminus \Sigma_\square^T$ and $\mathcal{L}_E : E \rightarrow \mathbb{I}(\mathbb{N}) \times \Sigma_\square^T$ (we denote the projection of \mathcal{L}_E on the first component by $\mathcal{L}_E|_I$);
- for each root-to-leaf path $\pi = (v_0, v'_0) \dots (v_n, v'_n)$ in \mathcal{T} , $\rho_\pi = \mathcal{L}_E(v_0, v'_0)(\mathcal{L}_E|_I(v_0, v'_0), \mathcal{L}_V(v'_0)) \dots \mathcal{L}_E(v_n, v'_n)(\mathcal{L}_E|_I(v_n, v'_n), \mathcal{L}_V(v'_n))$ is a successful admissible run for φ (hereafter, we denote the set of all runs ρ_π associated with root-to-leaf paths in \mathcal{T} by $\text{runs}_{\mathcal{T}}$);
- for each run ρ_\square in \square -proj, there is a path $\pi = (v_0, v'_0) \dots (v_n, v'_n)$ such that $\rho_\square[0 \dots n] = \mathcal{L}_E(v_0, v'_0) \dots \mathcal{L}_E(v_n, v'_n)$.

It can be easily checked that a Σ_\square^T -response tree encodes some (successful) Σ_\square^T -response strategy \mathcal{S}_\square . Then, checking whether a formula φ and a set $\Sigma_\square^T \subseteq \Sigma^T$ are a positive instance of the finite synthesis problem amounts to check whether there exists a Σ_\square^T -response tree for φ .

Theorem 6. *Let φ be an $\text{AB}\bar{\text{B}} \sim$ formula and $\Sigma_\square^T \subseteq \Sigma^T$. Then, $(\varphi, \Sigma_\square^T)$ is a positive instance of the finite-synthesis problem if and only if there exists a Σ_\square^T -response tree \mathcal{T} for φ .*

Unfortunately, this is not the end of the story. Every Σ_\square^T -response tree $\mathcal{T} = (\mathcal{T}, \mathcal{L}_\pi)$ is finite by definition, but this is not sufficient to conclude that the finite synthesis problem is decidable. To this end, we must provide a bound on the height of the tree depending on the size of φ . Let $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$ be a compass structure. For all $x, y \in D$, with $x \leq y$, we define the *multi-set of atoms* $M(x, y) = \{F : \exists x' \in D(x' \leq y \wedge x' \sim x \wedge \tau(x', y) = F)\}$, where the number of copies of F in $M(x, y)$, denoted by $|M(x, y)(F)|$, is equal to $|\{x' \in D : x' \leq y \wedge x' \sim x \wedge \tau(x', y) = F\}|$. Moreover, for all $y \in D$, we define the multiset-collection $\mathcal{M}(y)$ as the multi-set of multi-sets of atoms such that, for each multiset of atoms M , $\mathcal{M}(y)(M) = |\{[x]_\sim : x \in D \wedge x \leq y \wedge M(x, y) = M\}|$. Finally, we define a partial order \leq over the set of all multi-set collections as follows: for any pair of multi-set collections $\mathcal{M}, \mathcal{M}'$, $\mathcal{M} \leq \mathcal{M}'$ if and only if there exists an injective multi-set function $g \subseteq \mathcal{M} \times \mathcal{M}'$ such that, for each pair $(M, M') \in g$, $M \subseteq M'$ (an injective function $g \subseteq \mathcal{M} \times \mathcal{M}'$ between two multisets is itself a multiset such that $g|_1 = \mathcal{M}$ and $g|_2 \subseteq \mathcal{M}'$, where $|_i$ is simply the projection on the i -th component of a tuple). The following property of \leq is not difficult to prove, but it is crucial for the decidability proof.

Lemma 2. \leq is a well-quasi-ordering (WQO) over multiset collections.

Let $\mathcal{G} = (\mathbb{D} \times \mathbb{D}, \tau)$ be a compass structure for φ . We say that \mathcal{G} is *minimal* if and only if for all $y < y'$ in D , $\mathcal{M}(y) \not\leq \mathcal{M}(y')$. Given an $\text{ABB} \sim$ formula φ and a Σ_{\square}^T -response tree \mathcal{T} for it, we say that \mathcal{T} is *minimal* if and only if for each $\rho \in \text{runs}_{\mathcal{T}}$, $\mathcal{G}_{\rho_{\mathbb{M}}}$ is a minimal compass structure. The following result allows us to restrict our attention to minimal Σ_{\square}^T -response trees.

Lemma 3. Let φ be an $\text{ABB} \sim$ formula and Σ_{\square}^T be a finite set of its variables. Then, if there exists a Σ_{\square}^T -response tree \mathcal{T} for φ , then there exists a minimal Σ_{\square}^T -response tree \mathcal{T}' for φ .

Proof. (sketch) Let $\mathcal{T} = (V, E, \mathcal{L}_V, \mathcal{L}_E)$ be a Σ_{\square}^T -response tree for φ . Suppose that \mathcal{T} is not minimal. We show that there exists a smaller Σ_{\square}^T -response tree for φ which can be obtained by contracting one among the paths of \mathcal{T} that violate minimality. Notice that, in doing that, we prove that any defect (with respect to minimality) can be fixed by reducing the size of the tree in such a way that the resulting tree is still a Σ_{\square}^T -response tree for φ .

Since \mathcal{T} is not minimal, there is a run $\rho \in \text{runs}_{\mathcal{T}}$ such that $\mathcal{G}_{\rho_{\mathbb{M}}} = (\mathbb{D} \times \mathbb{D}, \tau)$ is not a minimal compass structure. Then, there are $y < y'$ in D such that $\mathcal{M}(y) \leq \mathcal{M}(y')$. Let $g \subseteq \mathcal{M} \times \mathcal{M}'$ be the function from $\mathcal{M}(y)$ to $\mathcal{M}(y')$, whose existence is guaranteed by definition of \leq . By the definition of the collections, injectivity of g implies the existence of an injective function $f : \{0, \dots, y\} \rightarrow \{0, \dots, y'\}$ such that $\tau(x, y) = \tau(f(x), y')$ and for each pair $0 \leq x \leq x' \leq y$, $x \sim x'$ if and only if $f(x) \sim f(x')$. Let $\pi = (v_0, v'_0) \dots (v_n, v'_n)$ be a root-to-leaf path such that $\rho = \mathcal{L}_E(v_0, v'_0), (\mathcal{L}_E|_I(v_0, v'_0), \mathcal{L}_V(v'_0)) \dots \mathcal{L}_E(v_n, v'_n) (\mathcal{L}_E|_I(v_n, v'_n), \mathcal{L}_V(v'_n))$ (the existence of such a path is guaranteed by the definition of Σ_{\square} -response tree). Given a node $v \in V$, we denote by $\mathcal{T}_v = (V_v, E_v, \mathcal{L}_{E_v}, \mathcal{L}_{V_v})$ the sub-tree of \mathcal{T} rooted at v . Let v_i (resp., v_j) be a node in π such that i (resp., j) is the minimum index for which the interval $[x, y''] = \mathcal{L}_E|_I(v_i, v'_{i+1})$ (resp., $[x, y''] = \mathcal{L}_E|_I(v_j, v'_{j+1})$) satisfies $y'' > y$ (resp., $y'' > y'$), for all $i' \geq i$ (resp., $j' \geq j$).

Let $V'_{v_j} = \{v \in V_{v_j} : \exists \pi' = (v_0, v'_0) \dots (v_m, v'_m)$ in \mathcal{T}_{v_j} s.t. $v_0 = v_j \wedge v'_m = v \wedge \forall 0 \leq i \leq m$ if $\mathcal{L}_{E_{v_j}}(v_i, v'_i)|_I = [x, y'']$ then $x > y' \vee x \in \text{Img}(f)\}$. The set V'_{v_j} collects all and only the nodes reachable from v_j through a path of edges which feature only intervals $[x, y'']$ such that either $x \geq y'$ (that is, the point has been introduced after y') or $x \in \text{Img}(f)$. Moreover, let $E'_{v_j} = \{(v, v') \in E_{v_j} : v, v' \in V'_{v_j} \cup \{v_j\}\}$ be the set of edges restricted to the set V'_{v_j} and let $\mathcal{L}'_{V_{v_j}}(v) = \mathcal{L}_{V_{v_j}}(v)$ for all $v \in V'_{v_j}$. Finally, let $\Delta = y' - y$. We define $\mathcal{L}'_{E_{v_j}}$ in such a way that, for each $(v, v') \in E'_{v_j}$, if $(\mathcal{L}_{E_{v_j}}(v, v') =) \mathcal{L}_E(v, v') = ([x, y''], \sigma_{\square})$, then $\mathcal{L}'_{E_{v_j}}(v, v') = ([x', y' - \Delta], \sigma_{\square})$, where $x' = f^{-1}(x)$ if $x \leq y'$ (in such a case, by construction, $x \in \text{Img}(f)$) and, since f is injective, it can be inverted on x) or $x' = x - \Delta$ otherwise.

We complete the construction by replacing (in \mathcal{T}) the subtree \mathcal{T}_{v_i} by the subtree $\mathcal{T}'_{v_j} = (V'_{v_j} \cup \{v_i\}, E'_{v_j} \setminus \{(v_j, v) \in E\} \cup \{(v_i, v) : \exists (v_j, v) \in E\}, \mathcal{L}'_{V_{v_j}}, \mathcal{L}'_{E_{v_j}})$, where $\mathcal{L}'_{V_{v_j}}(v) = \mathcal{L}_{V_{v_j}}(v)$ for each $v \in V'_{v_j}$, $\mathcal{L}'_{V_{v_j}}(v_i) = \mathcal{L}_V(v_i)$, $\mathcal{L}'_{E_{v_j}}(v, v') = \mathcal{L}'_{E_{v_j}}(v, v')$ for each $(v, v') \in E'_{v_j}$ with $v, v' \in V'_{v_j}$, and $\mathcal{L}'_{E_{v_j}}(v_i, v) = \mathcal{L}'_{E_{v_j}}(v_j, v)$ for each $(v_j, v) \in E'_{v_j}$ with $v \in V'_{v_j}$. It is possible to prove (by induction) that the tree \mathcal{T}' obtained from such a contraction operation on subtrees is still a Σ_{\square}^T -response tree for φ . \square

This proof provides the necessary insights for devising a decision procedure to establish whether or not $(\varphi, \Sigma_{\square})$ is a positive instance of the finite synthesis problem. Such a procedure visits a (candidate) Σ_{\square} -response tree \mathcal{T} for φ in a breadth-first fashion. At each step, the number of total edges from level l to level $l + 1$ of the tree is finite, and thus their labeling \mathcal{L}_E as well as labeling \mathcal{L}_V for the nodes at level $l + 1$ can be nondeterministically guessed. The procedure returns success if it finds a Σ_{\square} -response tree for φ ; it returns failure (that is, the generated candidate tree is not minimal) if it either introduces some local inconsistency or it produces a path such that there exist two coordinates $y_i < y_j$ with $\mathcal{M}(y_i) \leq \mathcal{M}(y_j)$. Being \leq is a WQO guarantees that a path cannot be arbitrarily long.

Theorem 7. *The finite synthesis problem for $ABB \sim$ is decidable.*

6 Conclusion

In this paper, we explored the synthesis problem for meaningful fragments of *HS* in the presence of an equivalence relation over points. On the negative side, we proved that the computational complexity of the synthesis problem is generally worse than that of the corresponding satisfiability problem (from elementary/decidable to nonelementary/undecidable). On the positive side, we showed that the increase in expressiveness makes it possible to capture new interesting temporal conditions.

References

- [1] J. Allen (1983): *Maintaining Knowledge about Temporal Intervals*. *Communications of the ACM* 26(11), pp. 832–843, doi:10.1145/182.358434.
- [2] J. R. Büchi & L. H. Landweber (1990): *Solving Sequential Conditions by Finite-State Strategies*. In S. Mac Lane & D. Siefkes, editors: *The Collected Works of J. R. Büchi*, Springer New York, pp. 525–541, doi:10.2307/1994916.
- [3] A. Church (1957): *Applications of recursive arithmetic to the problem of circuit synthesis*. In: *Summaries of the Summer Institute of Symbolic Logic, Volume I*, Cornell Univ., Ithaca, N.Y., pp. 3–50, doi:10.2307/2271310.
- [4] T. French, J. C. McCabe-Dansted & M. Reynolds (2012): *Synthesis for Temporal Logic over the Reals*. In: *Proc. of the 9th AiML*, pp. 217–238.
- [5] J. Halpern & Y. Shoham (1991): *A propositional modal logic of time intervals*. *Journal of the ACM* 38(4), pp. 935–962, doi:10.1145/115234.115351.
- [6] Z. Manna & P. Wolper (1984): *Synthesis of Communicating Processes from Temporal Logic Specifications*. *ACM Transactions on Programming Languages and Systems* 6(1), pp. 68–93, doi:10.1145/357233.357237.
- [7] M. L. Minsky (1967): *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., NJ, USA.
- [8] A. Montanari, G. Puppis & P. Sala (2010): *Maximal decidable fragments of Halpern and Shoham’s modal logic of intervals*. In: *Proc. of the 37th ICALP - Part II, LNCS 6199*, pp. 345–356, doi:10.1007/978-3-642-14162-1_29.
- [9] A. Montanari, G. Puppis, P. Sala & G. Sciavicco (2010): *Decidability of the Interval Temporal Logic ABB on Natural Numbers*. In: *Proc. of the 27th STACS*, pp. 597–608, doi:10.4230/LIPIcs.STACS.2010.2488.
- [10] A. Montanari & P. Sala (2013): *Adding an Equivalence Relation to the Interval Logic ABB : Complexity and Expressiveness*. In: *Proc. of the 28th LICS, IEEE*, pp. 193–202, doi:10.1109/LICS.2013.25.
- [11] A. Pnueli & R. Rosner (1989): *On the Synthesis of a Reactive Module*. In: *Proc. of the 16th POPL*, ACM Press, pp. 179–190, doi:10.1145/75277.75293.
- [12] A. Rabinovich (2007): *The Church Synthesis Problem with Parameters*. *Logical Methods in Computer Science* 3(4), doi:10.2168/lmcs-3(4:9)2007.
- [13] A. Rabinovich & W. Thomas: *Logical refinements of Church’s Problem*. In: *Proc. of the 21st CSL, LNCS 4646*, Springer, pp. 69–83, doi:10.1007/978-3-540-74915-8_9.
- [14] P. Schnoebelen (2010): *Lossy Counter Machines Decidability Cheat Sheet*. In A. Kucera & I. Potapov, editors: *Reachability Problems, LNCS 6227*, Springer, pp. 51–75, doi:10.1007/978-3-642-15349-5_4.
- [15] Y. Venema (1991): *A modal logic for chopping intervals*. *Journal of Logic and Computation* 1(4), pp. 453–476, doi:10.1093/logcom/1.4.453.